Amendments to the Claims:

This listing of claims will replace all prior versions and listings of claims in the application:

Listing of Claims:

1. (currently amended) An electronic device circuit comprising:

a bus interface for communications with a host;

an interface unit electrically coupled to the bus interface for, in a startup procedure, receiving operational firmware from the host and receiving initialization data required for initializing the electronic device from the host;

a control circuit electrically coupled to the interface unit for transferring the received operational firmware to a volatile memory; and

a microprocessor electrically coupled to the control circuit for executing the received operational firmware while stored in the volatile memory;

wherein the microprocessor controls ~~the normal operations of~~ the electronic device circuit according to the received operational firmware, and the electronic device circuit is initialized by the initialization data which is received in the startup procedure ~~contains instructions required to initialize the components of the electronic device circuit before the microprocessor is able to execute the operational firmware~~.

2. (previously presented) The electronic device circuit of claim 1 wherein the bus interface conforms to USB, IDE, SATA, SAS, or SCSI interface standards.

3. (previously presented) The electronic device circuit of claim 1 wherein the interface unit is a macro.

4. (previously presented) The electronic device circuit of claim 3 wherein the macro comprises handshaking, data reception, and writing received data into the memory functions.

5-6. (cancelled)

7. (previously presented) The electronic device circuit of claim 1 wherein the host is a computer system.

8. (previously presented) The electronic device circuit of claim 1 wherein the microprocessor executes the received operational firmware without accessing a non-volatile memory.

9. (cancelled)

10. (currently amended) The electronic device circuit of claim 1 wherein the volatile memory comprises the received operational firmware being executed by the microprocessor to control ~~normal operations of~~ the electronic device circuit.

11. (currently amended) An electronic device comprising a download mode wherein, in a startup procedure, operational firmware is received from an external host and stored into a volatile memory of the electronic device and initialization data required for initializing the electronic device is received from the external host, the electronic device is initialized by the initialization data received in the startup procedure, followed by a normal mode wherein a microprocessor of the electronic device executes the operational firmware received in the startup procedure ~~stored in the volatile memory~~ to control ~~normal operations of~~ the electronic device~~, wherein the initialization data contains instructions required to initialize the components of the electronic device before the microprocessor is able to execute the operational firmware~~.

12. (currently amended) The electronic device of claim 11 wherein ~~the normal operations of~~ the electronic device at least ~~include reading~~ reads data from ~~an~~

~~optical disc~~ a storage medium, processing the data, and transferring the processed data to the host.

13. (cancelled)

14. (previously presented) The electronic device of claim 11 wherein the operational firmware is received over a bus interface conforming to USB, IDE, SATA, SAS, or SCSI interface standards.

15. (previously presented) The electronic device of claim 11 wherein the host is a computer system.

16. (currently amended) A method of operating an electronic device, the electronic device comprising a control circuit connected to a microprocessor, a volatile memory, and a bus interface connected to a host, the method comprising:

receiving operational firmware from the host in a startup procedure;

receiving initialization data ~~required for initializing the electronic device~~ from the host when the electronic device in the [[a]] startup procedure, wherein the initialization data contains instructions required to initialize ~~the components of~~ the electronic device ~~before the microprocessor is able to execute the operational firmware~~;

initializing the electronic device by the initialization data which is received in the startup procedure;

writing the operational firmware into the volatile memory; and

~~the microprocessor~~ executing the operational firmware in the volatile memory by the microprocessor to control ~~normal operations of~~ the electronic device.

17. (cancelled)

18. (previously presented) The method of claim 16 wherein the operational firmware is received over a bus interface conforming to USB, IDE, SATA, SAS, or SCSI interface standards.

19. (previously presented) The method of claim 16 further comprising the electronic device transmitting an electrical signal to an application program in the host to begin receiving the operational firmware.

20. (original) The method of claim 16 wherein the host is a computer system.

21. (currently amended) A computer system comprising:

a host computer comprising operational firmware for controlling operations of an electronic device and initialization data for initializing the electronic device; and

the electronic device comprising:

a volatile memory ~~comprising~~ for storing the operational firmware transferred ~~received~~ from the host computer through a connecting bus interface; and

a microprocessor executing the operational firmware stored in the volatile memory for controlling ~~normal operations of~~ the electronic device;

wherein the electronic device further receives the initialization data from the host computer when the electronic device is in a startup procedure, ~~the initialization data contains instructions required to initialize the components of the electronic device before the microprocessor is able to execute the operational firmware, the components of the electronic device being initialized only after the electronic device receives the initialization data from the host computer~~ and the electronic device is initialized by the initialization data received in the startup procedure.

22. (currently amended) The computer system of claim 21 wherein the microprocessor

executes the operational firmware to control ~~the normal operations of~~ the electronic device ~~at least include controlling the rotational speed of an optical disc in the electronic device and reading~~ to read data from a storage medium ~~the optical disc~~.

23. (original) The computer system of claim 21 wherein the bus interface conforms to USB, IDE, SATA, SAS, or SCSI interface standards.

24. (cancelled)

25. (currently amended) An electronic device controller comprising:

a bus interface for communications with a host;

a volatile memory for storing operational firmware received from the host in a startup procedure;

a microprocessor for controlling ~~normal operations of~~ the electronic device by executing the operational firmware stored in the volatile memory;

an RF circuit; and

a control circuit connected to the bus interface, the volatile memory, the microprocessor, and the RF circuit;

wherein the electronic device controller is initialized by initialization data ~~required for initializing the electronic device is~~ received from the host in the [[a]] startup procedure, ~~the initialization data contains instructions required to initialize the components of the electronic device before the microprocessor is able to execute the operational firmware~~.

26. (currently amended) The electronic device controller of claim 25 wherein the volatile memory comprises the received operational firmware being executed by the microprocessor to control ~~normal operations of~~ the electronic device.

27. (currently amended) An electronic device circuit used in a host system, wherein

the electronic device circuit has operational firmware transferred from the host system to a volatile memory through a bus interface after the host being powered on, the electronic device circuit comprising:

a microprocessor for executing the received operational firmware while stored in the volatile memory;

wherein the electronic device circuit is initialized by ~~the microprocessor controls the normal operations of the electronic device according to the received operational firmware, and~~ initialization data ~~required for initializing the electronic device circuit~~ which is ~~received~~ transferred from the host system after the host being powered on ~~in a startup procedure, the initialization data contains instructions required to initialize the components of the electronic device circuit before the microprocessor is able to execute the operational firmware~~.

28. (previously presented) The electronic device circuit of claim 27 wherein the bus interface conforms to USB, IDE, SATA, SAS, or SCSI interface standards.

29. (cancelled)

30. (previously presented) The electronic device circuit of claim 27 wherein the host system is a computer system.

31. (previously presented) The electronic device circuit of claim 27 wherein the microprocessor executes the received operational firmware without accessing a non-volatile memory.

32. (previously presented) The electronic device circuit of claim 27 wherein the host system comprises the volatile memory.

33. (currently amended) The electronic device circuit of claim 27 wherein the host

system comprises a host controller accessing the volatile memory that is shared by the host system and the microprocessor ~~during the normal operation~~.

34. (currently amended) The electronic device circuit of claim 27 wherein the volatile memory is accessed only by the electronic device circuit ~~on the normal operation~~.

35. (previously presented) The electronic device circuit of claim 27 wherein the electronic device circuit comprises the volatile memory.

36-37. (cancelled)